Functional Activity Based Comparison Study for Neural Network Application

Ahmad A. Al-Rababah[†] **& Mohammad A. Al-Rababah**[††]

[†] Faculty of IT, Amman Al-Ahliyyah University, P.O.Box 252, 19328 Amman, Jordan.
[††]Faculty of Science, CS Department, Jerash Private University,Jordan.

Summary

. This paper presents brief study mainly focused on the modeling aspect of the transfer function of the primitive element of a neuron. Then, the transfer function is drawn for an investigation towards its analysis and classification. This analysis has proposed nine different cases of proper modes.

The paper from another view of study discusses the main attributes needed in neural structures to yield sufficient correlation between the activation function classification and the different aspects of structural view of the networking purposes. The correlation had been decided on the bases of the resulted outcomes of conducted experiments.

1. Introduction

The elementary structural concept of neural network design adopts the processing element of a neuron as a basic computation unit. Along the progressive developments of neural network researches, this processing element had been supported by the necessary capabilities to interpret the analog behavior that had generalized the abstraction of the digital responses of the earlier models.[1,2]

The sought emergent need for all the trends of the enhancement actually is considered as to upgrade the transfer function to draw it within the convenience of sophisticated applications. Definitely, any advancement carried out is heavily invoking a suitable learning algorithm that is required to set up the weighting scheme. [3]

Recent past studies have investigated the effect of the transfer function on the association process that attributes any output with its related input with a minimum error.[5,2] The major effort in adopting any new behavior or characteristic of the transfer function of a neuron denotes the learning that is attributed to such new function. In this context different works had been presented to evolve the convergence criterion of the learning algorithm. Besides, there are other characteristics that been investigated including generalization, memory size capacity of associations covered in a given architecture. []1,3,7]

The current work studies the effect of the activation on the different characteristics of neuron network performance these characteristics are focused on the learning phase.[4,5,9] The study is based on investigating the different modes of the activation function in order to classify their features. The work resumes to study the major fields of application. Then a correction process between to activation function from a side and the application field from the other side is decided.[8,10]

Manuscript received January 5, 2007
Manuscript revised January 25, 2007

2. Formal Neuron Model

As it is mentioned, a formal neuron is the main element in NN structure. It executes the parametric non-linear transformation of input vector X into scalar quantity Y. This transformation has two stages: 1 – The front end stage calculates, the discriminate function which is truncated multidimensional Taylor series; 2 – it's transformed into output quality Y. Coefficients of Taylor series form the vector of weight coefficient w, or the neuron memory. The discriminate function of the first order neuron is [2].

The discrete mode includes the form

$$\Psi(x)=\begin{cases}1, & x\geq 0\\ 0, & x<0\end{cases}, \tag{1}$$

$$\Psi(x)=\begin{cases}+1, & x\geq 0\\ -1, & x<0\end{cases}, \tag{2}$$

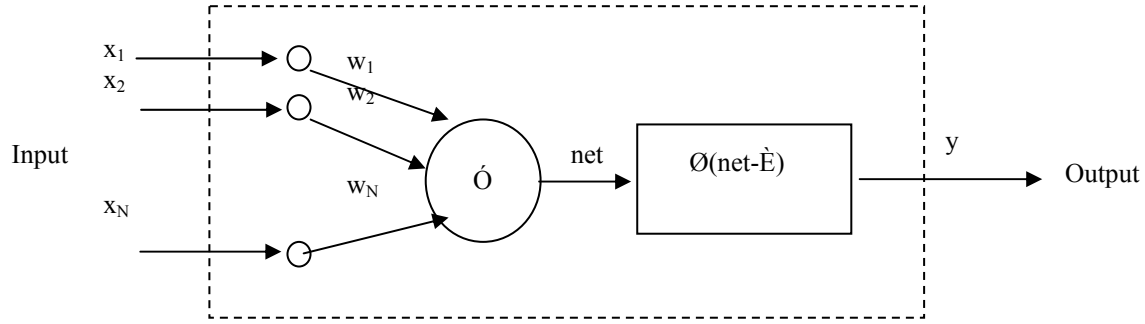

Figure 1 - Formal Neuron Modeling

$$net=\sum_{i=1}^{N} w_i x_i,$$

where N, is the number of inputs xi and wi , is neuron weight coefficients. Fig.1. demonstrates the modeling aspect in to a suitable scheme representing a formal neuron.

3. Activation Function Classification

T The overall transfer function of a neuron that is given by $y=\Psi(net-\Theta)$ represents a non-linear transformation that is characterized by Ψ for which Θ is considered as the moderating threshold. In this modeling Ψ Could be classified to fall into two different modes.

a. Discrete mode, and
b. Classical Analog mode.

Whereas the second mode combines all the modalities of the sigmoid function as

$$\Psi(x)=\frac{1}{1+e^{-x}}. \tag{3}$$

The second mode depicts all the functions that is organized to have the sequences of the positive terms and the negative ones. However, this mode could include a wide range of functions that range in their convenience for possible implementation in activation function modeling.

The applyment of any activation function should support the characteristic that satisfies the overall association and thus to result in the following cases: -

Case one

The activation function has to be bounded i.e. y^0 has to be obtained in interval (-∞, à], y1 in interval [b,+∞) and $y^0 \leq y \leq y^1$ in the interval (a , b),

where y^0 and y^1 represent some constants of a minimum and a maximum values respectively. Usually those values are o and one or -1 and +1 correspondingly. In general a $\leq$ b.

Case two

The transformation has to be monotonic in the interval (a,b), i.e. $\Delta\psi(\tilde{o})= \psi(\tilde{o}+\Delta\tilde{o})- \psi(\tilde{o})$ dose not change the sign at $\Delta\tilde{o}>0$/ and $\tilde{o} \in (a,b)$, $x + \Delta x \in (a,b)$.

For cases 1 and 2 , it is obvious that there are infinite set of functions that could be formulated but for proper implementation of these characteristics, it is emergent to imply the third case:-

Case three

Alteration function has to e easily and quickly calculated, i.e. it has to contain as small as possible of arithmetic operations deduced by the given constants and simple operations. These operations would be further approved for their convenience when are considered as library function in any computation tool, along the implemented utility for programming purposes.

By considering the above cases, it is thus possible to extend the first three modes given in (1) to (3), to involve additional six ones. These functions are:

$$\Psi(x)=\begin{cases}1, & x>1\\ \dfrac{x^n+1}{2}, & -1\le x\le 1,\\ 0, & x<-1\end{cases} \quad \text{where n is the twin number}, \quad n>0 \tag{4}$$

$$\Psi(x)=\begin{cases}1, & x>\dfrac{\sigma}{2}\\ \dfrac{\sin(x)+1}{2}, & -\dfrac{\sigma}{2}\le x\le\dfrac{\sigma}{2},\\ 0, & x<-\dfrac{\sigma}{2}\end{cases} \tag{5}$$

$$\Psi(x)=\begin{cases}1-e^{-xx}/2, & x\ge 0\\ \dfrac{e^{-xx}}{2}, & x<0\end{cases}, \tag{6}$$

$$\Psi(x)=\frac{\tanh(x)+1}{2}, \quad \textit{where} \quad \tanh(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}. \tag{7}$$

$$\Psi(x)=\begin{cases}1, & x>1\\ \dfrac{\arcsin(x)+1}{2}, & -1\le x\le 1,\\ 0, & x<-1\end{cases} \tag{8}$$

$$\Psi(x)=\begin{cases}1, & x>\dfrac{\sigma}{4}\\ \dfrac{tg(x)+1}{2}, & -\dfrac{\sigma}{4}\le x\le\dfrac{\sigma}{4},\\ 0, & x<-\dfrac{\sigma}{4}\end{cases} \tag{9}$$

When neural network arguments are considered, there are another view that characterizes the applyment matter of the proper activation function needed. This view demonstrates the constraints of the built up. The following sub sections discuss the different architectures along the emergent features of the activation functions.

4. Neural Network Considerations

4.1 One Layer Perception

A Perception represents a formal neuron. The output generated from such an architecture is either a discrete or an analogue ((real)) therefore, it could be decided that a class of **case four** is involved. This condition addresses a discrete one layer Perceptron that describes an activation function with the only a value y^0 or y^1 . This may use only threshold functions of (1) or (2).

Case five

describes a one-layer perception with a real output that may use any function of (3) to (9) with the exemption of implementing (1) and (2).

4.2 Multi Layers Neural Networks (MNNs)

This class represents the network, which consists of some formal neurons that are connected in the feed forward mode, i.e. every neuron is connected to a next layer neurons but not in its layer. MNN with discrete outputs imposes the same activation function requirements of that of a one-layer Perceptron case 4 for discrete mode and of that of case 5 for real mode.

The algorithms of MNN training impose additional restrictions on the activation function that are summarized by the following cases.

Case six

Gradient method of training or what so called inverse spreading of mistake uses the correcting rule of scales $w_{k+1}= w_k- a_k g_k$ where w_k , is the running vector of weights and thresholds of NN, g_k is the running gradient, a_k is the velocity of training. This method imposes the activation function, which has the same characteristics of case 5 and also it requires that this function should be differentiable with the condition that the first derivative of y (x) would not be identically equal to zero, other wise the activation function would be constant.

Case seven

Newton's method or the method of inverse spreading of mistake of the second order uses the correcting rule of the form $w_{k+1}= w_k-À_k^{-1}g_k$, where A^{-1}_k is the matrix of the second order derivatives

This methods imposes the same requirements of that of case six in addition to the condition that the second derivative should not be identically equal zero.

In this case, the algorithm of the conjugate gradients proposed by Fletcher Rives, could be involved. This algorithm uses the correcting rule of masses, given by
$p_0=-g_0$, $w_{k+1}=w_k+a_kp_k$, $p_{k+1}=-g_k+b_kp_{k-1}$, $â_k =(g_k^Tg_k)/(g_{k-1}^Tg_{k-1})$. In addition, the algorithm proposed by Pollak Reabyer is also attributed to the same condition of case seven. This algorithm uses the correcting mass rule of the following $p_k=-g_k+b_kp_{k-1}$, $â_k =(\Delta g_{k-1}^Tg_k)/(g_{k-1}^Tg_{k-1})$. Finally this case involves also the algorithm of Lerenberg- Markvardt that uses the following rule:-
$H=J^TJ$, $g=J^Te$, $w_{k+1}=w_k+[H_k+mI]^{-1}g_k$, where J is Jakobian, e – vector of mistakes, I- unit matrix

4. Case Investigation and Comparison Methodologies

The forgoing discussion had overviewed the different cases in most of the requirements of the structures and the algorithms. For such discussion, it could be concluded that for the cases of discrete models, only activation functions of (1) and (2) are applicable. Where as, for the cases of NN of the real mode, (3) to (9) satisfy the requirements of the concrete models and algorithms of the related training.

The main goal for the present work is focused on the decision of fitting an application with a proper activation function, besides, the criteria on which this decision is made. However, this goal adopts the single layer Perceptron structure and MNN, for the investigation, as they are widely being used in the different fields of application. Generally speaking, both of the structures could use any activation function given in cases (3) to (9), but essentially for any decision, two questions arise for such decision.

1. What functions of (3) to (9) are considered as the best ones.
2. What are the criteria that ought to be taken in to consideration for the decision making.

From the foregoing description of the training algorithms of NN, it follows, that convergence of training algorithms depends on:-

Functio	Mean Computing Time (1/1000	Mean time of training at fixed maximal number of training cycles (1000 cycles),se		Mean value of trainin ycle
		On -layer perception, Algorithm of Widro -Hoff	MN -Algorithm of Levenberg-Markvard	
3	2,09	0,77	4,61	4,7
4	0,61	0,67	3,5	3,3
5	1,0	0,78	3,57	3,3
6	1,94	0,78	3,71	3,7
7	3,66	0,93	3,53	3,0
8	1,24	0,31	3,91	4,7
9	0,71	0,84	3,79	4,0

1. Criterion of training quality that denotes the range of the permissible error of trimming.
2. Criterion of training speeds that denotes the number of permissible cycles of training.

Definitely, the above criterions stand besides the training data and initial weight factors.

It is expediently to use the above two criteria for the comparison process of activation functions of (3) to (9). In such concern, it is possible to observe, how for each NN model, the value of another criterion is changing at the same set of the different training samples with the same initial weightings. For which, it is necessary to appraise the experimental results depending on the criterions. These criterions are interpreted in terms of operating time of training algorithm and the training cycles.

6. Experimentation

The experimentation conducted had utilized a common utility for all the tested architectures and the rule for the learning phase (training) purposes. The experiments are applied on one layer Perceptron, two and three layers MNN. The experiments had involved the testing of different input layout as well. In these experiments, pseudorandom numbers and samples of real product parameters are used. The outcome of the conducted experiments are summarized in Table -1-

7. Results and conclusions

From Table -1-, the most remarks that summarize the experimentation and the overall comparison yields the following results:-

1. Functions of cases (3) to (5) in addition to (8) for the one layer Perceprion model in most cases allows to achieve convergence at

fixed minimal common error much more quickly than the other functions.

2. Functions (3), (6) and (7) and because of the fast exponent increase, result in malfunction resulted from the overflow of word size.
3. Functions (3), (5), (6) and (8) at fixed maximum number of training cycles (10,100,1000) give less common error than what the other functions give.
4. Functions (4) to (6) besides (7) give the possibility to carryout the necessary convergence for relatively a smaller number of training cycles.

Summing up all of the above remarks, two final conclusion remarks are extracting be major advice: -

- For one layer Perceptron, the most proper activation functions are (3) to (5) besides (8).
- For MNN, the most proper activation functions are, (4), (5) and (7).

References

1. DAN W. PATERSON, 1996, Artificial Neural Networks, Theory and Applications.
2. MOHAMMAD H. HASSOUN, 1998, Fundamentals of Artificial Neural Networks.
3. BART KOSKO, 1997, Neural Networks and Fuzzy Systems, A Dynamical Systems Approach to machine Intelligence.
4. BART KOSKO, 1991, Neural Networks for Signal processing.
5. CARDOT H. REVEN4 M., VICTORRI B., REVILLET J., 1993, An Artificial Networks Architecture for Handwritten Signature Authentication, Proceeding of the SPIE Applications of Artificial Neural Networks II, Orlando, FL.
6. CARPENTER G., GROSSBERG S., 1987, A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine, Computer Vision, Graphics, and Image Processing, vol. 37, pp. 54-115.
7. WIDROW B., WINTER R., 1988, Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition, IEEE Computer, vol. 21, No.3, PP. 25-39.
8. KOSKO B., Neural Networks for signal processing, prentice Hall, Englewood Cliffs, NJ, 1991.
9. KONG S. G., KOSKO B., "Differential Competitive Learning for Censored Estimation and Phoneme Recognition" IEEE Transaction on Neural Network, Vol.2, No.1, 118-124, January, 1991.
10. KOSKO B. "Unsupervised learning in Noise," IEEE Transaction of Neural Networks, Vol.1, No.1, 44-57, March 1990.

Ahmad A. Al-Rababah is a staff member of Faculty of Information Technology, Amman Ahliyyah University- Jordan. He obtained his MSc & PhD in computer engineering from USSR in the area of information technology and control systems. He is currently being IT committee member that is recommended through the Quality Assurance requirements in 2002-2006. His interest areas are: Artificial Neural Network, Web Designing , Distributed and Control systems.

Mohammad A. Al-Rababah is an assistant professor, Jerash private University- Jordan. He obtained his MSc & PhD in computer engineering from USSR in the area of computer network security. His interest areas are: Networks, Artificial Intelligence, Quality Assurance requirements